Electronic Colloquium on Computational Complexity, Report No. 12 (2015)

Lower Bounds for Clique vs. Independent Set

Mika Göös

Department of Computer Science, University of Toronto

January 24, 2015

Abstract

We prove an $\omega(\log n)$ lower bound on the conondeterministic communication complexity of the Clique vs. Independent Set problem introduced by Yannakakis (STOC 1988, JCSS 1991). As a corollary, this implies superpolynomial lower bounds for the Alon–Saks–Seymour conjecture in graph theory. Our approach is to first exhibit a query complexity separation for the decision tree analogue of the UP vs. coNP question—namely, unambiguous DNF width vs. CNF width—and then "lift" this separation over to communication complexity using a result from prior work.

1 Introduction

Yannakakis's [Yan91] *Clique vs. Independent Set* problem, associated with an undirected n-node graph $G = ([n], E)$, is the following: Alice holds a clique $x \subseteq [n]$ in G, Bob holds an independent set $y \subseteq [n]$ in G, and their goal is to decide whether x and y intersect. As the underlying graph enforces $|x \cap y| \in \{0, 1\}$, we may define a boolean function by $\mathrm{CIS}_G(x, y) := |x \cap y|$.

Upper bounds. For every G there is an $\lceil \log n \rceil$-bit nondeterministic communication protocol for CIS_G that guesses the name of the unique node in the intersection $x \cap y$. Recall (e.g., [KN97, Juk12]) that, combinatorially, this means that the 1-entries of the communication matrix of CIS_G can be covered with n rectangles. We write this fact as $\mathsf{NP}^{\mathsf{cc}}(\mathrm{CIS}_G) \leq \lceil \log n \rceil$. Yannakakis further proved that $\mathsf{P}^{\mathsf{cc}}(\mathrm{CIS}_G) \leq O(\log^2 n)$, where P^{cc} stands for deterministic communication complexity. In particular, we have the same upper bound on the conondeterministic complexity:

$$\mathsf{coNP}^{\mathsf{cc}}(\mathrm{CIS}_G) \;\leq\; O(\log^2 n).$$

Lower bounds. It has been a relatively long-standing open problem to prove (for some choice of G) superlogarithmic lower bounds on $\mathsf{P}^{\mathsf{cc}}(\mathrm{CIS}_G)$, let alone on $\mathsf{coNP}^{\mathsf{cc}}(\mathrm{CIS}_G)$. See, for instance, the textbooks by Kushilevitz and Nisan [KN97, Exercise 1.8] and Jukna [Juk12, Research Problem 4.15]. See also Table 1 for a summary of previous bounds, and the works of Kushilevitz and Weinreb [KW09a, KW09b] for indirect attacks on the problem.

Our main result is to obtain such superlogarithmic lower bounds.

Theorem 1. *There is a family of graphs G such that*

$$\mathsf{coNP}^{\mathsf{cc}}(\mathrm{CIS}_G) \;\geq\; \Omega(\log^{1.128} n).$$

ISSN 1433-8092

Measure	Lower bound	Reference
P^{cc}	$2 \cdot \log n$	Kushilevitz, Linial, and Ostrovsky [KLO99]
$\mathsf{coNP}^{\mathsf{cc}}$	$6/5 \cdot \log n$	Huang and Sudakov [HS12]
$\mathsf{coNP}^{\mathsf{cc}}$	$3/2 \cdot \log n$	Amano [Ama14]
$\mathsf{coNP}^{\mathsf{cc}}$	$2 \cdot \log n$	Shigeta and Amano [SA14]
$\mathsf{coNP}^{\mathsf{cc}}$	$\omega(\log n)$	This work

Table 1: Lower bounds on the deterministic (P^{cc}) and conondeterministic ($\mathsf{coNP}^{\mathsf{cc}}$) communication complexities of the Clique vs. Independent Set problem.

Implications. The CIS problem admits several equivalent formulations, as explored in-depth by Bousquet, Lagoutte, and Thomassé [BLT14]. In particular, Theorem 1 refutes a certain "polynomial" version of the Alon–Saks–Seymour conjecture [BLT14, Conjecture 16]. The original conjecture stated that $\chi(G) \leq \mathrm{bp}(G) + 1$, that is, that the chromatic number of G can be bounded in terms of the *biclique packing number* of G (minimum number of complete bipartite graphs needed to partition the edges of G). Huang and Sudakov [HS12] disproved the original conjecture by showing that $\chi(G)$ can be polynomially larger than $\mathrm{bp}(G)$. Theorem 1 implies that the gap can be superpolynomial. See [BLT14] for more details about this and other connections.

1.1 Our approach

Unambiguity. The canonical $\lceil \log n \rceil$-cost nondeterministic protocol for CIS_G outlined above has the additional property of being *unambiguous*: on each input the protocol can accept at most one nondeterministic guess. That is, combinatorially, the rectangles covering the 1-entries do not overlap. Thus $\lceil \log n \rceil$ is an upper bound on the unambiguous communication complexity of CIS_G, which we write as $\mathsf{UP}^{\mathsf{cc}}(\mathrm{CIS}_G) \leq \lceil \log n \rceil$. In fact, it is known that the CIS_G family of problems is *complete* for unambiguous communication: for every two-party function $F \colon \mathcal{X} \times \mathcal{Y} \to \{0,1\}$ there is a graph G on $n = 2^{\mathsf{UP}^{\mathsf{cc}}(F)}$ nodes such that F appears as a subproblem of CIS_G. See Figure 1 for an illustration. In particular, we have $\mathsf{UP}^{\mathsf{cc}}(F) = \mathsf{UP}^{\mathsf{cc}}(\mathrm{CIS}_G) = \log n$. Given this structural perspective, our goal becomes to exhibit a total[1] function with a large $\mathsf{UP}^{\mathsf{cc}}$ vs. $\mathsf{coNP}^{\mathsf{cc}}$ gap.

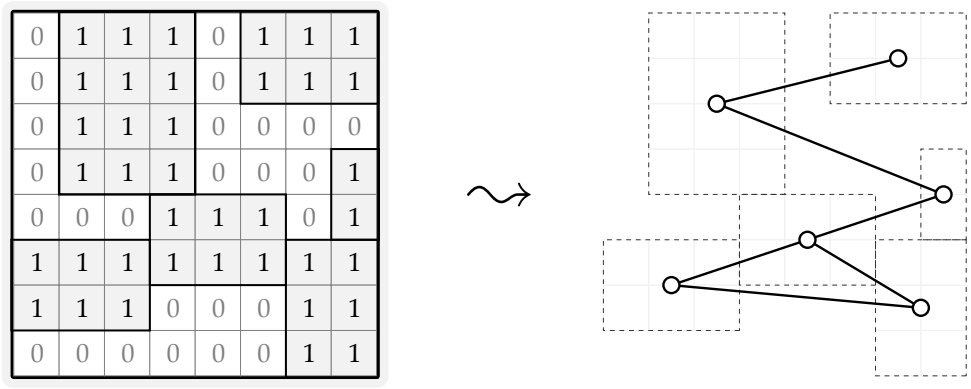
The following is a rephrasing of Theorem 1.

Theorem 2. *There is an F such that $\mathsf{coNP}^{\mathsf{cc}}(F) \geq \mathsf{UP}^{\mathsf{cc}}(F)^{\beta}$ where $\beta > 1.128$ is a constant.*

Query complexity. Instead of attacking Theorem 2 head-on, we first show an analogous separation in the simpler-to-understand world of decision tree complexity [BdW02]. Here one deals with plain boolean functions $f \colon \{0,1\}^n \to \{0,1\}$ (different n than above) without any two-party structure. We use the superscript "dt" for query complexity measures. For example, $\mathsf{NP}^{\mathsf{dt}}(f)$ denotes the nondeterministic decision tree complexity of f, or equivalently, the minimum k such that f can be written as a k-DNF. We also set $\mathsf{coNP}^{\mathsf{dt}}(f) \coloneqq \mathsf{NP}^{\mathsf{dt}}(\neg f)$. A DNF is *unambiguous* if on every input at most one of its terms (a.k.a. certificates) evaluates to true. We define $\mathsf{UP}^{\mathsf{dt}}(f)$ as the minimum k such that f can be written as an unambiguous k-DNF. (Unambiguous decision trees have been studied at least in [Sav02].)

The following is the query analogue of Theorem 2.

[1] It is easy to give a partial function (promise problem) with an exponential $\mathsf{UP}^{\mathsf{cc}}$ vs. $\mathsf{coNP}^{\mathsf{cc}}$ gap: the *unique set-intersection* function UINTER satisfies $\mathsf{UP}^{\mathsf{cc}}(\mathrm{UINTER}) \leq O(\log n)$ but $\mathsf{coNP}^{\mathsf{cc}}(\mathrm{UINTER}) \geq \Omega(n)$ [Raz92, KW14].



Partition of $F^{-1}(1)$ Graph G

Figure 1: Reduction $F \leq \mathrm{CIS}_G$ [Yan91, Lemma 1]: Fix an optimal partition of the 1-inputs of F using $2^{\mathsf{UP}^{\mathsf{cc}}(F)}$ rectangles. The nodes of the graph G are the rectangles, and two nodes are connected by an edge if the corresponding rectangles share a row. We have $F \leq \mathrm{CIS}_G$ via the following map: Alice maps her input x to the set of all rectangles intersecting row x, and Bob maps his input y to the set of all rectangles intersecting column y.

Theorem 3. *There is an f such that $\mathsf{coNP}^{\mathsf{dt}}(f) \geq \mathsf{UP}^{\mathsf{dt}}(f)^{\alpha}$ where $\alpha > 1.128$ is a constant.*

From query to communication. Given an f as above, we then apply a result from [GLM+14] that allows us to convert f into a communication problem while preserving its conondeterministic complexity. Specifically, we consider a composed function of the form $F := f \circ g^n$ where $g\colon \{0,1\}^b \times \{0,1\}^b \to \{0,1\}$ is a two-party "gadget" on a small number $b = \Theta(\log n)$ of inputs. The inputs to F are pairs $(x, y) \in \{0,1\}^{bn} \times \{0,1\}^{bn}$, which we view as being partitioned into n disjoint blocks $x_1, \ldots, x_n$ and $y_1, \ldots, y_n$ of b bits each. The task is to compute

$$F(x, y) \;:=\; f(g(x_1, y_1), \ldots, g(x_n, y_n)).$$

We use the following special case of a result proved in [GLM+14]—the full result there holds also for many randomised models. (We include a proof of this special case in Appendix A for completeness.)

Theorem 4 ([GLM+14])**.** *There is a gadget g on $b = \Theta(\log n)$ bits so that for all $f\colon \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{coNP}^{\mathsf{cc}}(f \circ g^n) \;\geq\; \Omega(\mathsf{coNP}^{\mathsf{dt}}(f) \cdot b). \tag{1}$$

This establishes the lower bound on $\mathsf{coNP}^{\mathsf{cc}}(F)$. For the upper bound on $\mathsf{UP}^{\mathsf{cc}}(F)$, we use the simple fact that a protocol can always simulate a corresponding type of decision tree:

$$\mathsf{UP}^{\mathsf{cc}}(f \circ g^n) \;\leq\; O(\mathsf{UP}^{\mathsf{dt}}(f) \cdot b). \tag{2}$$

Indeed, when a nondeterministic decision tree queries the i-th input of f, we simulate this query in a nondeterministic protocol by a proof string of length $\Theta(b) = \Theta(\log n)$ that contains an encoding of i ($\lceil \log n \rceil$ bits), Alice's i-th input x_i (b bits), and Bob's i-th input y_i (b bits). Given this information, both players can learn the output $g(x_i, y_i)$ of the i-th gadget. Finally, it is straightforward to check that the resulting protocol is unambiguous iff the decision tree is unambiguous.

To conclude, Theorem 3 together with (1) and (2) imply a gap of $\mathsf{coNP}^{\mathsf{cc}}(F) \geq \Omega(\mathsf{UP}^{\mathsf{cc}}(F)^{\alpha} \cdot \log^{1-\alpha} n)$. Our F is actually going to satisfy $\mathsf{UP}^{\mathsf{cc}}(F) \geq n^{\Omega(1)}$ so the previous bound can be written as $\mathsf{coNP}^{\mathsf{cc}}(F) \geq \mathsf{UP}^{\mathsf{cc}}(F)^{\alpha - o(1)}$. This proves Theorem 2 for any $\beta < \alpha$.

The rest of this paper is devoted to proving Theorem 3.

2 Definitions

A size-k *certificate* for "$P(x)$"—where $P(\cdot)$ is a predicate and $x \in \{0,1\}^n$ is an input—is a partial assignment C to inputs, setting at most k variables, such that x agrees with C (we also say that C accepts x) and for every y agreeing with C it holds that $P(y)$. The *certificate complexity* of "$P(x)$" is the least size of a certificate for "$P(x)$". A nondeterministic ($\mathsf{NP}^{\mathsf{dt}}$) decision tree $\mathcal{T}$ of cost k for "$P(\cdot)$" is a collection of size-k certificates such that for every x, $P(x)$ holds iff there exists an $C \in \mathcal{T}$ that accepts x. We say that $\mathcal{T}$ is *unambiguous* ($\mathsf{UP}^{\mathsf{dt}}$), if on every input x there is at most one certificate $C \in \mathcal{T}$ that accepts x. We define $\mathsf{NP}^{\mathsf{dt}}(f)$ (resp. $\mathsf{UP}^{\mathsf{dt}}(f)$) as the least cost of a $\mathsf{NP}^{\mathsf{dt}}$ (resp. $\mathsf{UP}^{\mathsf{dt}}$) decision tree for "$f(\cdot) = 1$". Also let $\mathsf{coNP}^{\mathsf{dt}}(f) \coloneqq \mathsf{NP}^{\mathsf{dt}}(\neg f)$. Finally, $\mathsf{P}^{\mathsf{dt}}(f)$ denotes the deterministic decision tree complexity of f.

3 Warm-up discussion

The purpose of this section is to develop a "feel" for the $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ question, and motivate some of the choices that are made in the upcoming sections.

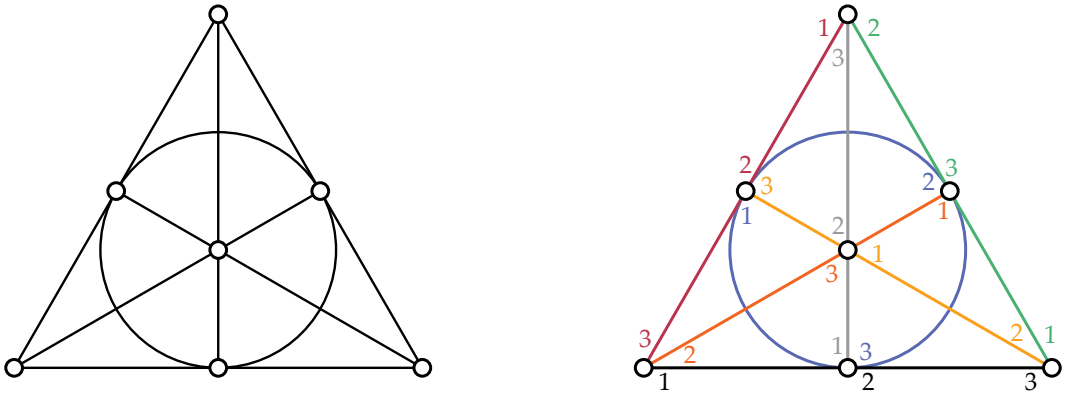
Upper bound on gap A well-known result for decision trees states that $\mathsf{P}^{\mathsf{dt}}(f) \leq \mathsf{NP}^{\mathsf{dt}}(f) \cdot \mathsf{coNP}^{\mathsf{dt}}(f)$; see [Juk12, §14.2]. A similar argument shows that $\mathsf{P}^{\mathsf{dt}}(f) \leq \mathsf{UP}^{\mathsf{dt}}(f)^2$. In particular, $\mathsf{coNP}^{\mathsf{dt}}(f) \leq \mathsf{UP}^{\mathsf{dt}}(f)^2$, and hence the $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ gap cannot be too large (as in communication complexity). This argument is illuminating, so we present it here.

Proposition 5. $\mathsf{P}^{\mathsf{dt}}(f) \leq \mathsf{UP}^{\mathsf{dt}}(f)^2$.

Proof. Let $\mathcal{T}$ be a $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f. A key observation is that any two certificates of $\mathcal{T}$ must intersect in variables—otherwise we could construct an input that satisfied two distinct certificates of $\mathcal{T}$, which contradicts unambiguity. The P^{dt} decision tree is this: Choose any certificate $C \in \mathcal{T}$ and query all its variables (at most $\mathsf{UP}^{\mathsf{dt}}(f)$ many). Let ρ be a partial assignment recording the values revealed, and let f_ρ be the restriction of f by ρ. If f_ρ is constant, we can output the corresponding value. Otherwise recursively run a decision tree for f_ρ. Here $\mathsf{UP}^{\mathsf{dt}}(f_\rho) \leq \mathsf{UP}^{\mathsf{dt}}(f) - 1$ as every remaining certificate $C' \neq C$ of $\mathcal{T}$ must intersect C in variables, so ρ already fixes at least one variable from each C'. □

Recursive composition It is easy to exhibit a constant-size function with *some* $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ gap. For example, define a function on 3 bits by $f(x) = 1$ iff x has Hamming weight 1 or 2. We have $\mathsf{UP}^{\mathsf{dt}}(f) = 2$ since $\{x_1\bar{x}_2,\ x_2\bar{x}_3,\ x_3\bar{x}_1\}$ is an unambiguous collection of certificates for f. We also have $\mathsf{coNP}^{\mathsf{dt}}(f) = 3$ since the function is sensitive to flipping any of the bits on the all-0 input $\vec{0}$.

A standard trick in boolean function complexity [NS94, NW95, Sav02] is to take a constant-size function, such as f above, and to recursively compose it with itself: $f^{i+1}(\cdot) \coloneqq f(f^i(\cdot), f^i(\cdot), f^i(\cdot)))$. The hope is that at every level of recursion the gap between the two complexity measures of interest increases by some constant factor (e.g., 3/2 in the case of f). However, this approach works straightforwardly only when the complexity measures are two-sided, that is, they do not distinguish between f and $\neg f$. In our case of $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ we are searching for a function whose 1-inputs are easy, but 0-inputs are hard. An obstacle associated with a recursively defined function f^i is that the only way to certify "$f^i(\cdot) = 1$"—which should be easy—often involves recursively certifying that "$f^{i-1}(\cdot) = 0$"—which should be hard!



(a) (b)

Figure 2: (a) Fano plane with 7 nodes and 7 hyperedges, and (b) a symmetric incidence ordering.

Large alphabets Our construction will employ recursion. In order to avoid the obstacle described above, we are going to enlarge the input/output alphabets of our functions. With large alphabets we assume a decision tree model where queries are still atomic, and reading a symbol from any input costs one query. We will study functions of the form $f\colon (\{0\} \cup \Sigma)^n \to \{0\} \cup \Sigma$, where the symbols in Σ are intuitively thought of as "easy to certify". The symbol 0 will be "hard to certify", and consequently we will always study the $\mathsf{coNP}^{\mathsf{dt}}$ complexity relative to the all-0 input $\vec{0}$. The idea is that certifying "$f^i(\,\cdot\,) = \sigma$" for $\sigma \in \Sigma$ is easy as we only need to recursively certify the same type of statements: "$f^{i-1}(\,\cdot\,) = \sigma'$" for $\sigma' \in \Sigma$. In particular, we make sure that an $\mathsf{UP}^{\mathsf{dt}}$ decision tree will never need to certify statements of the form "$f^i(\,\cdot\,) = 0$". Such $\mathsf{UP}^{\mathsf{dt}}$ decision trees will be called 0-*avoiding* later.

Note that if we have a function $f\colon \Sigma^n \to \{0,1\}$ with a large input alphabet, we may always convert it to a boolean function by composing it with n copies of some surjection $g\colon \{0,1\}^{\lceil \log |\Sigma| \rceil} \to \Sigma$. The following naïve bounds will suffice for our purposes:

$$\mathcal{C}(f) \;\le\; \mathcal{C}(f \circ g^n) \;\le\; \mathcal{C}(f) \cdot \lceil \log |\Sigma| \rceil \qquad \text{for all } \; \mathcal{C} \in \{\mathsf{UP}^{\mathsf{dt}}, \mathsf{coNP}^{\mathsf{dt}}\}. \tag{3}$$

4 Projective plane example

In this section we describe an example witnessing a constant-factor $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ gap. The example is based on projective planes, and it will be suitable for recursive composition later.

Finite projective planes. For our purposes, a finite projective plane H is a k-uniform hypergraph with $n = k^2 - k + 1$ nodes and n hyperedges. Each node is incident to k edges, and each edge is incident to k nodes. Moreover, the edges are pairwise intersecting. See Figure 2a for an example of the case $k = 3$. It is well known that finite projective planes exist whenever $k - 1$ is a prime power.

Symmetric incidence ordering. Given an H as above, we fix for each node $v \in V(H)$ an ordering of its incident edges, and for each edge $e \in E(H)$ an ordering of its incident nodes. We make sure that the two orderings are *symmetric*: e is the i-th incident edge of $v \in e$ iff v is the i-th incident node of e; see Figure 2b. Such symmetric orderings are guaranteed to exist by Hall's

matching theorem [Die10, §2.1]. (Consider the k-regular bipartite incidence graph of H that has the original nodes $V(H)$ on the left and the original hyperedges $E(H)$ on the right and there is an edge $\{v, e\}$ iff $v \in e$. Then by Hall's theorem the edges of this graph can be partitioned into k disjoint perfect matchings—this encodes the desired orderings.)

Unweighted function. To turn H into a query problem f, we make the nodes of H correspond to input variables that take on values from $\{0\} \cup [k]$. These values are interpreted as *pointers*: we say that a node with input $i \in [k]$ *points* to its i-th incident edge. A node with input 0 does not point anywhere (0 is a *null pointer*). Moreover, we say that an assignment $x\colon V(H) \to \{0\} \cup [k]$ *satisfies* an edge $e \in E(H)$ if all the incident nodes $v \in e$ point to e according to x. Note that each x can satisfy at most a single edge, since every two edges share a node, and this node can only point to one of the two edges.

The function $f\colon (\{0\} \cup [k])^n \to \{0, 1\}$ is now defined so that $f(x) = 1$ iff x satisfies an edge. There is an obvious cost-k $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f whose certificates are in one-to-one correspondence with the edges of H. Unfortunately, the certificate complexity of "$f(\vec{0}) = 0$" is not large: it is also k. We will next fix this by assigning *weights* to the inputs.

Input weights. We are going to modify the function f defined above so that querying the inputs x_j to f becomes harder: to decide whether "$x_j = i$" one needs to spend $w(i)$ queries (instead of 1) where $w\colon [k] \to \mathbb{N}$ is a *weighting* function of our choosing. We allow only positive weights, so we agree that $0 \notin \mathbb{N}$. Concretely, this is achieved by considering a composed function $f \circ g_w^n$ where g_w is a gadget that implements the weights w. More specifically, g_w is of the form $(\{0\} \cup [k])^m \to \{0\} \cup [k]$ where $m \coloneqq \max w([k])$ is the maximum weight and $g_w(x)$ is defined as follows: If $x = \vec{0}$, output 0. Otherwise let $x_j = i$ be the first non-0 coordinate of x. If $j \leq w(i)$, output i; otherwise output 0.

By construction, for deterministic decision trees and for every $i \in [k]$, it is both necessary and sufficient to make $w(i)$ queries in order to decide whether "$g_w(\,\cdot\,) = i$". Furthermore, we record for later use the following two properties that also follow straightforwardly from the construction.

(P1) The certificate complexity of "$g_w(\vec{0}) \neq i$" is $w(i)$.
(P2) Suppose $\hat{w}(i) \coloneqq \ell \cdot w(i)$ for all i. Then $\mathsf{coNP}^{\mathsf{dt}}(f_{\hat{w}}) = \ell \cdot \mathsf{coNP}^{\mathsf{dt}}(f_w)$ for every f.

Weighted function. Define w by $w(i) \coloneqq i$ and consider $f_w \coloneqq f \circ g_w^n$. We claim that

$$\begin{aligned} \mathsf{UP}^{\mathsf{dt}}(f_w) &\leq k(k+1)/2, \\ \mathsf{coNP}^{\mathsf{dt}}(f_w) &\geq k^2 - k + 1. \end{aligned}$$

That is, we have asymptotically a factor 2 separation.

Upper bound. We can devise a $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f_w by simply composing a $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f and optimal deterministic decision trees for the g_w's. A certificate corresponding to an edge $e \in V(H)$ in the $\mathsf{UP}^{\mathsf{dt}}$ decision tree for "$f(\,\cdot\,) = 1$" checks for each $i \in [k]$ that the i-th incident node of e is outputting i, which recursively involves checking that "$g_w(\,\cdot\,) = i$" for the corresponding gadget. For each i this amounts to $w(i) = i$ queries, which is $\sum_{i \in [k]} i = k(k+1)/2$ in total.

Lower bound. Our goal is to lower bound $\mathsf{coNP}^{\mathsf{dt}}(f_w)$ by analysing the 0-input $\vec{0}$. We start by highlighting a special property of the gadget that is afforded by our choice of w. The property states that to certify—on input $\vec{0}$—that none of a set of ℓ pointers appear as the output of g_w, one needs to make at least ℓ queries to the input variables.

Claim 6. *Let $S \subseteq [k]$. The certificate complexity of "$g_w(\vec{0}) \notin S$" is at least $|S|$.*

Proof. To certify "$g_w(\vec{0}) \notin S$" it is necessary to certify "$g_w(\vec{0}) \neq i$" where i is the largest number in S. But this latter task needs certificates of size $w(i) = i \geq |S|$ by *(P1)*. □

Our special property actually holds more generally across all the n gadgets. This generalised property states that to certify—on input $\vec{0}$—that none of a set of ℓ pointers (possibly associated with different nodes of H) appear in the output of the n gadgets g_w^n, one needs to make at least ℓ queries to the input variables of g_w^n. To state this more precisely, write $G \coloneqq g_w^n$ and let $G_v(\,\cdot\,)$ denote the output of the gadget corresponding to node $v \in V(H)$.

Claim 7. *For each v let $S_v \subseteq [k]$. The certificate complexity of "$\forall v : G_v(\vec{0}) \notin S_v$" is at least $\sum_v |S_v|$.*

Proof. Claim 6 proves this for each fixed v, but since the gadgets are defined on disjoint sets of variables, the individual certificate complexities sum up. □

Finally, the lower bound follows from the fact that any certificate for "$f_w(\vec{0}) = 0$" must certify that each edge $e \in E(H)$ lacks at least one pointer. This is $n = k^2 - k + 1$ pointers (and hence queries) in total.

5 Recursive composition

In this section we prove Theorem 3, restated below for convenience. In short, our idea is to recursively compose the example of Section 4.

Theorem 3. *There is an f such that $\mathsf{coNP}^{\mathsf{dt}}(f) \geq \mathsf{UP}^{\mathsf{dt}}(f)^{\alpha}$ where $\alpha > 1.128$ is a constant.*

Conventions. In what follows, we will consider pairs (f, w) where $f\colon (\{0\} \cup \Sigma)^n \to \{0,1\}$ is a function and $w\colon \Sigma \to \mathbb{N}$ assigns weights to the inputs of f. We also define $f_w \coloneqq f \circ g_w^n$ as before. A 0-*avoiding* $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f is one whose certificates contain only values from Σ (i.e., not 0). More generally, a 0-*avoiding* $\mathsf{UP}^{\mathsf{dt}}$ decision tree for a weighted function f_w is a composition of a 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f and a deterministic decision tree for the g_w's; here a decision tree for g_w can—and indeed sometimes must—read 0's from the inputs. We let $\mathsf{UP}^{\mathsf{dt}}_\star(f_w)$ stand for the minimum cost of a 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree for f_w. We prove our $\mathsf{coNP}^{\mathsf{dt}}$ lower bound by considering the complexity of certifying "$f_w(\vec{0}) = 0$", that is, we only focus on the 0-input $\vec{0}$. Hence we introduce the notation $\mathsf{coNP}^{\mathsf{dt}}_\star(f_w)$ to stand for the certificate complexity of "$f_w(\vec{0}) = 0$".

Overview. Given a pair (f, w), we construct a new pair (f', w') with the following properties. (Here k is again a parameter, chosen later, such that $k-1$ is a prime power.)

(A1) Unambiguous complexity: $\mathsf{UP}^{\mathsf{dt}}_\star(f'_{w'}) \leq k(k+1)/2 \cdot \mathsf{UP}^{\mathsf{dt}}_\star(f_w)$.
(A2) Conondeterministic complexity: $\mathsf{coNP}^{\mathsf{dt}}_\star(f'_{w'}) \geq (k^2 - k + 1) \cdot \mathsf{coNP}^{\mathsf{dt}}_\star(f_w)$.

We proceed in two steps. In the first step (Section 5.1), we transform (f, w) into a pair $(\hat{f}, \hat{w})$ where $\hat{f}$ is a multi-valued function outputting values in $\{0\} \cup [k]$. The key property is the following:

> **If** f_w exhibits a gap between unambiguously certifying "$f_w(\,\cdot\,) = 1$" vs. nondeterministically certifying "$f_w(\vec{0}) = 0$", **then** for all $i \in [k]$, $\hat{f}_{\hat{w}}$ exhibits the same gap between unambiguously certifying "$\hat{f}_{\hat{w}}(\,\cdot\,) = i$" vs. nondeterministically certifying "$\hat{f}_{\hat{w}}(\vec{0}) \neq i$".

In the second step (Section 5.2), we begin thinking of $\hat{f}_{\hat{w}}$ as a gadget outputting pointer values: we plug several copies of $\hat{f}_{\hat{w}}$ as inputs to the projective plane example of Section 4. This will amplify the $\mathsf{UP}^{\mathsf{dt}}$ vs. $\mathsf{coNP}^{\mathsf{dt}}$ gap further and give us (f', w').

From this construction it is easy to derive Theorem 3 (Section 5.3).

5.1 First step: Multi-valued outputs

From $f\colon (\{0\} \cup \Sigma)^N \to \{0,1\}$ we will construct $\hat{f}\colon (\{0\} \cup \Sigma \times [k])^N \to \{0\} \cup [k]$. For notation, let $\pi_1\colon \Sigma \times [k] \to \Sigma$ and $\pi_2\colon \Sigma \times [k] \to [k]$ denote the natural projection maps, and extend $\pi_1(0) = \pi_2(0) = 0$ for convenience. For $x \in (\{0\} \cup \Sigma \times [k])^N$ we write $\pi_1(x)$ and $\pi_2(x)$ for the coordinate-wise application of π_1 and π_2 to x.

Definition of $(\hat{f}, \hat{w})$. Fix some optimal 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree $\mathcal{T}$ for f. We define $\hat{f}$ on input $x \in (\{0\} \cup \Sigma \times [k])^N$ as follows: If $f(\pi_1(x)) = 0$, output 0. Otherwise consider the unique certificate of $\mathcal{T}$ that accepts $\pi_1(x)$. This certificate reads some subset $S \subseteq [N]$ of inputs, where $x_j \neq 0$ for all $j \in S$, since $\mathcal{T}$ is 0-avoiding. If there is an $i \in [k]$ such that $\pi_2(x_j) = i$ for all $j \in S$, then output i; otherwise output 0.

The input weights $\hat{w}\colon \Sigma \times [k] \to \mathbb{N}$ are defined by $\hat{w}(\sigma, i) \coloneqq w(\sigma) \cdot i$.

Analysis. We claim that $(\hat{f}, \hat{w})$ satisfies the following.

(B1) There is a 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree for "$\hat{f}_{\hat{w}}(\,\cdot\,) = i$" with cost $i \cdot \mathsf{UP}^{\mathsf{dt}}_{\star}(f_w)$.

(B2) The certificate complexity of "$\hat{f}_{\hat{w}}(\vec{0}) \neq i$" is at least $i \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$.

(B1) holds: We can simply modify $\mathcal{T}$ slightly: to certify "$\hat{f}_{\hat{w}}(\,\cdot\,) = i$" the modified decision tree works as before but now additionally checks the new condition on $\pi_2(x)$ where x is the input to $\hat{f}$ (as encoded by $g_{\hat{w}}$). The cost of the modified tree is i times that of $\mathcal{T}$ as the relevant inputs are now i times heavier in the new $\hat{w}$ than in the old w.

(B2) holds: We prove this by a reduction argument. Fix $i \in [k]$. Consider deleting all symbols from the input alphabet of $\hat{f}$ except $\{0\} \cup \Sigma \times \{i\}$ and call the resulting subfunction $\hat{\hat{f}}$. Clearly the certificate complexity of "$\hat{f}_{\hat{w}}(\vec{0}) \neq i$" (which we are interested in) is at least that of "$\hat{\hat{f}}_{\hat{w}}(\vec{0}) \neq i$". Note that the range of $\hat{\hat{f}}$ is just $\{0, i\}$. Moreover, $\hat{\hat{f}}$ is isomorphic to f: identify the input alphabets via the map π_1, and the output alphabets via the map $0 \mapsto 0$, $i \mapsto 1$. Therefore the certificate complexities of "$\hat{\hat{f}}_{\hat{w}}(\vec{0}) \neq i$" and "$f_{\hat{w}}(\vec{0}) = 0$" are the same. (Here we abused notation by identifying the alphabets of f and $\hat{w}$ according to the isomorphism.) But $\hat{w}$ is nothing but w scaled by a factor of i (on the relevant alphabet $\Sigma \times \{i\}$), so by *(P2)* the certificate complexity of "$f_{\hat{w}}(\vec{0}) = 0$" is i times that of "$f_w(\vec{0}) = 0$", namely $i \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$.

5.2 Second step: Composition with a projective plane

Take a k-uniform projective plane hypergraph H as in Section 4 and let its associated unweighted function be $h\colon (\{0\} \cup [k])^n \to \{0,1\}$ where $n = k^2 - k + 1$. We define $f' \coloneqq h \circ \hat{f}^n$ together with the old weights $w' \coloneqq \hat{w}$ (so $f'_{w'} = h \circ \hat{f}^n_{\hat{w}}$). We claim that (f', w') satisfies *(A1)* and *(A2)*.

(A1) holds: We do the natural thing: The 0-avoiding $\mathsf{UP}^{\mathsf{dt}}$ decision tree for $f'_{w'}$ has a certificate for every edge $e \in E(H)$ that, for every $i \in [k]$, recursively checks that the i-th node incident to e points to e. The i-th recursive check involves certifying "$\hat{f}_{\hat{w}}(\,\cdot\,) = i$", which we can do in a 0-avoiding manner using *(B1)*. This has total cost $\sum_{i \in [k]} i \cdot \mathsf{UP}^{\mathsf{dt}}_{\star}(f_w) = k(k+1)/2 \cdot \mathsf{UP}^{\mathsf{dt}}_{\star}(f_w)$, as desired.

(A2) holds: We now begin thinking of the $\hat{f}_{\hat{w}}$'s as "gadgets". Under this nomenclature, we can simply repeat the argument from Section 4 verbatim. For example, our special property in the case of a single gadget states the following: to certify—on input $\vec{0}$—that none of a set of ℓ pointers appear as the output of $\hat{f}_{\hat{w}}$, one needs to make at least $\ell \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$ queries to the inputs of $\hat{f}_{\hat{w}}$.

Claim 8. *Let $S \subseteq [k]$. The certificate complexity of "$\hat{f}_{\hat{w}}(\vec{0}) \notin S$" is at least $|S| \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$.*

Proof. To certify "$\hat{f}_{\hat{w}}(\vec{0}) \notin S$" it is necessary to certify "$\hat{f}_{\hat{w}}(\vec{0}) \neq i$" where i is the largest number in S. But this latter task needs certificates of size $i \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w) \geq |S| \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$ by *(B2)*. □

As before, because the n gadgets $\hat{f}^n_{\hat{w}}$ are fed disjoint sets of variables, a more general property holds: to certify—on input $\vec{0}$—that none of a set of ℓ pointers (possibly associated with different nodes of H) appear in the output of the n gadgets $\hat{f}^n_{\hat{w}}$, one needs to make at least $\ell \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$ queries to the input variables of $\hat{f}^n_{\hat{w}}$.

Finally, the lower bound follows from the fact that any certificate for "$f'_{w'}(\vec{0}) = 0$" must certify that each edge $e \in E(H)$ lacks at least one pointer. This is $n = k^2 - k + 1$ pointers that require $(k^2 - k + 1) \cdot \mathsf{coNP}^{\mathsf{dt}}_{\star}(f_w)$ queries in total.

5.3 Putting everything together

Define a sequence of pairs (f^i, w^i) as follows: Initially, $f^0 \colon \{0,1\} \to \{0,1\}$ is the identity function and $w^0(1) \coloneqq 1$. Clearly the $\mathsf{UP}^{\mathsf{dt}}_{\star}$ and $\mathsf{coNP}^{\mathsf{dt}}_{\star}$ complexities for $f^0_{w^0}$ are both 1. Recursively define (f^{i+1}, w^{i+1}) as the "primed" versions of (f^i, w^i).

By inspecting the construction, we have the following properties.

- Number of inputs to f^i is $(k^2 - k + 1)^i$.
- Maximum weight of w^i is k^i.
- Therefore, the number of inputs to $f^i_{w^i}$ is $n \coloneqq (k^3 - k^2 + k)^i$.
- Input alphabet $\{0\} \cup \Sigma$ of $f^i_{w^i}$ has size $|\{0\} \cup \Sigma| = 1 + k^i$.
- *(A1)* implies: $\mathsf{UP}^{\mathsf{dt}}(f^i_{w^i}) \leq (k(k+1)/2)^i$.
- *(A2)* implies: $\mathsf{coNP}^{\mathsf{dt}}(f^i_{w^i}) \geq (k^2 - k + 1)^i$.

The last two items say that $\mathsf{coNP}^{\mathsf{dt}}(f^i_{w^i}) \geq \mathsf{UP}^{\mathsf{dt}}(f^i_{w^i})^\beta$ for $\beta \coloneqq \log(k^2 - k + 1)/\log(k(k+1)/2)$. This is maximised at $k = 8$ (among k such that $k - 1$ is a prime power), which yields $\beta = \log_{36} 57 > 1.128$.

As a final step, we need to replace the input alphabet of $f^i_{w^i}$ with a binary encoding. The above parameters reveal that $\log |\{0\} \cup \Sigma| \leq O(\log n)$ while our estimates for both $\mathsf{coNP}^{\mathsf{dt}}(f^i_{w^i})$ and $\mathsf{UP}^{\mathsf{dt}}(f^i_{w^i})$ are $n^{\Theta(1)}$. Hence the loose bounds (3) give us the gap in Theorem 3 for any $\alpha < \beta$.

A Appendix: Proof of Theorem 4

In this appendix—in order to make this work self-contained—we reproduce the proof of Theorem 4 originally proved in [GLM+14] (as part of a more general result).

Theorem 4 ([GLM+14])**.** *There is a gadget g on $b = \Theta(\log n)$ bits so that for all $f \colon \{0,1\}^n \to \{0,1\}$,*

$$\mathsf{coNP}^{\mathsf{cc}}(f \circ g^n) \;\geq\; \Omega(\mathsf{coNP}^{\mathsf{dt}}(f) \cdot b). \tag{1}$$

We choose g to be the inner-product gadget on $b := 100 \log n$ bits per party:

$$g(x, y) := \langle x, y \rangle \bmod 2, \qquad \text{where } x, y \in \{0,1\}^b.$$

Write $G := g^n$ for short. It is useful to think of a rectangle $R = X \times Y$ (where $X, Y \subseteq \{0,1\}^{bn}$) via a pair of random variables $\boldsymbol{xy}$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ are uniformly (or otherwise) distributed on X and Y. A large rectangle corresponds to a pair $\boldsymbol{xy}$ where $\boldsymbol{x}$ and $\boldsymbol{y}$ have have high *min-entropy*, defined by $\mathbf{H}_\infty(\boldsymbol{x}) := \min_x \log(1/\mathbf{Pr}[\boldsymbol{x} = x])$ (see [Vad12] for more on min-entropy). We will use the fact that, if R is large, the output $\boldsymbol{z} := G(\boldsymbol{x}, \boldsymbol{y})$ has large support, that is, R accepts many different inputs to the outer function f. This is formalised in Lemma 9 below.

A.1 Density lemma

For $I \subseteq [n]$ and an input $x \in \{0,1\}^{bn}$, we write $x_I \in \{0,1\}^{b|I|}$ for the projection of x to the *blocks* indexed by I. We say that a pair $\boldsymbol{xy}$ of random variables is δ-*dense* if for all $I \subseteq [n]$ the blocks $\boldsymbol{x}_I\boldsymbol{y}_I$ have min-entropy rate at least δ, that is,

$$\mathbf{H}_\infty(\boldsymbol{x}_I\boldsymbol{y}_I) \geq \delta \cdot 2b|I|.$$

Lemma 9. *If $\boldsymbol{x}$ and $\boldsymbol{y}$ are independent and $\boldsymbol{xy}$ is 0.6-dense, then $\boldsymbol{z} := G(\boldsymbol{x}, \boldsymbol{y})$ has full support, namely $\{0,1\}^n$.*

Proof. [GLM+14, Lemma 13]. We actually prove a stronger claim: for $\epsilon = o(1)$, $\boldsymbol{z}$ is ϵ-*uniform*, meaning that $\mathbf{Pr}[\boldsymbol{z} = z] = (1 \pm \epsilon) \cdot 2^n$ for every $z \in \{0,1\}^n$. Clearly $\boldsymbol{z}$ has full support in this case.

First observe that for any $I \subseteq [n]$ the parity of the output bits $\boldsymbol{z}_I$ is simply $\langle \boldsymbol{x}_I, \boldsymbol{y}_I \rangle \bmod 2$. We use the fact that inner-product is a good two-source extractor to argue that this parity is close to an unbiased random bit. Indeed, by independence and 0.6-density we have $\mathbf{H}_\infty(\boldsymbol{x}_I) + \mathbf{H}_\infty(\boldsymbol{y}_I) = \mathbf{H}_\infty(\boldsymbol{x}_I\boldsymbol{y}_I) \geq 1.2 \cdot b|I|$ and this implies by a basic theorem of Chor and Goldreich [CG88, Theorem 9] that for $I \neq \emptyset$,

$$\left| \mathbf{Pr}[\langle \boldsymbol{x}_I, \boldsymbol{y}_I \rangle \bmod 2 = 0] - 1/2 \right| \leq 2^{-0.1 \cdot b|I|+1}. \tag{4}$$

This bound is enough to yield ϵ-uniformity for $\epsilon := 2^{-b/20}$, as we next verify using standard Fourier analysis. Let $\mathcal{D}$ be the distribution of $\boldsymbol{z}$. We think of $\mathcal{D}$ as a function $\{0,1\}^n \to [0,1]$ and write it in the Fourier basis as

$$\mathcal{D}(z) = \sum_{I \subseteq [n]} \widehat{\mathcal{D}}(I) \chi_I(z)$$

where $\chi_I(z) := (-1)^{\sum_{i \in I} z_i}$ and $\widehat{\mathcal{D}}(I) := 2^{-n} \sum_z \mathcal{D}(z)\chi_I(z) = 2^{-n} \cdot \mathbf{E}_{\boldsymbol{z} \sim \mathcal{D}}[\chi_I(\boldsymbol{z})]$. Note that $\widehat{\mathcal{D}}(\emptyset) = 2^{-n}$ because $\mathcal{D}$ is a distribution. In this language, property (4) says that, for all $I \neq \emptyset$, $2^n \cdot |\widehat{\mathcal{D}}(I)| = |\mathbf{E}[(-1)^{\langle \boldsymbol{x}_I, \boldsymbol{y}_I \rangle}]| \leq 2^{-0.1 \cdot b|I|+2}$, which is at most $\epsilon 2^{-2|I| \log n}$ by our definition of b and ϵ. Hence,

$$2^n \sum_{I \neq \emptyset} |\widehat{\mathcal{D}}(I)| \leq \epsilon \sum_{I \neq \emptyset} 2^{-2|I| \log n} = \epsilon \sum_{k=1}^{n} \binom{n}{k} 2^{-2k \log n} \leq \epsilon \sum_{k=1}^{n} 2^{-k \log n} \leq \epsilon.$$

We use this to show that $|\mathcal{D}(z) - 2^{-n}| \leq \epsilon 2^{-n}$ for all $z \in \{0,1\}^n$, which proves the lemma. To this end, let $\mathcal{U}$ denote the uniform distribution (note that $\widehat{\mathcal{U}}(I) = 0$ for all $I \neq \emptyset$) and let $\mathbb{1}_z$ denote the indicator for z defined by $\mathbb{1}_z(z) = 1$ and $\mathbb{1}_z(z') = 0$ for $z' \neq z$ (note that $|\widehat{\mathbb{1}}_z(I)| = 2^{-n}$ for all I). We can now calculate

$$\begin{aligned} |\mathcal{D}(z) - 2^{-n}| &= |\langle \mathbb{1}_z, \mathcal{D} \rangle - \langle \mathbb{1}_z, \mathcal{U} \rangle| = |\langle \mathbb{1}_z, \mathcal{D} - \mathcal{U} \rangle| = 2^n \cdot |\langle \widehat{\mathbb{1}}_z, \widehat{\mathcal{D}} - \widehat{\mathcal{U}} \rangle| \\ &\leq 2^n \cdot \textstyle\sum_{I \neq \emptyset} |\widehat{\mathbb{1}}_z(I)| \cdot |\widehat{\mathcal{D}}(I)| = \sum_{I \neq \emptyset} |\widehat{\mathcal{D}}(I)| \leq \epsilon 2^{-n}. \end{aligned}$$

□

A.2 Proof of Theorem 4

Overview. Let Π be a covering of the 0-inputs of $f \circ G$ using 2^c rectangles. Our goal will be to argue that for every $z \in f^{-1}(0)$ there is a small certificate for "$f(z) = 0$". More precisely, we are looking for a certificate C such that *(i)* C has size $O(c/b)$, *(ii)* C accepts z, and *(iii)* C only accepts 0-inputs of f. To find such a certificate, we find a subrectangle $R' \subseteq R$ of some $R \in \Pi$ such that $G(R')$ equals the set of inputs accepted by some C satisfying the properties prescribed above. More precisely, *(i)* $G(R')$ is a subcube of codimension $O(c/b)$, that is, $G(R')$ is fixed on $O(c/b)$ many coordinates and has full support elsewhere, *(ii)* $z \in G(R')$, and *(iii)* $G(R') \subseteq f^{-1}(0)$. Note that *any* $R' \subseteq R \in \Pi$ contains only 0-inputs of $f \circ G$ so that $G(R')$ contains only 0-inputs of f, that is, property *(iii)* will hold automatically.

Finding R'. Fix $z \in f^{-1}(0)$ and let $\boldsymbol{xy}$ be uniformly distributed on $G^{-1}(z)$. Using the fact that Π covers the whole support of $\boldsymbol{xy}$, we can find some $R \in \Pi$ such that $\mathbf{Pr}[\,\boldsymbol{xy} \in R\,] \geq 2^{-c}$. Denote by $(\boldsymbol{xy} \mid \boldsymbol{xy} \in R)$ the variables $\boldsymbol{xy}$ conditioned on the event "$\boldsymbol{xy} \in R$".

Let $I \subseteq [n]$ be a maximum-size subset for which 0.8-density is violated for $(\boldsymbol{xy} \mid \boldsymbol{xy} \in R)$, and let α be an outcome witnessing this: $\mathbf{Pr}[\,\boldsymbol{x}_I\boldsymbol{y}_I = \alpha \mid \boldsymbol{xy} \in R\,] > 2^{-1.6b|I|}$. We claim that conditioning further on the event "$\boldsymbol{x}_I\boldsymbol{y}_I = \alpha$", the remaining blocks indexed by $\bar{I} \coloneqq [n] \setminus I$ are 0.8-dense. Write $\boldsymbol{x}'\boldsymbol{y}' \coloneqq (\boldsymbol{xy} \mid \boldsymbol{xy} \in R,\ \boldsymbol{x}_I\boldsymbol{y}_I = \alpha)$ for short.

Claim 10. *$\boldsymbol{x}'_{\bar{I}}\boldsymbol{y}'_{\bar{I}}$ is 0.8-dense.*

Proof. Suppose not: there is some nonempty set $J \subseteq \bar{I}$ and a string β such that $\mathbf{Pr}[\,\boldsymbol{x}_J\boldsymbol{y}_J = \beta \mid \boldsymbol{xy} \in R,\ \boldsymbol{x}_I\boldsymbol{y}_I = \alpha\,] > 2^{-1.6b|J|}$. Now $\mathbf{Pr}[\,\boldsymbol{x}_I\boldsymbol{y}_I = \alpha \text{ and } \boldsymbol{x}_J\boldsymbol{y}_J = \beta \mid \boldsymbol{xy} \in R\,] = \mathbf{Pr}[\,\boldsymbol{x}_I\boldsymbol{y}_I = \alpha \mid \boldsymbol{xy} \in R\,] \cdot \mathbf{Pr}[\,\boldsymbol{x}_J\boldsymbol{y}_J = \beta \mid \boldsymbol{xy} \in R,\ \boldsymbol{x}_I\boldsymbol{y}_I = \alpha\,] > 2^{-1.6b|I|} \cdot 2^{-1.6b|J|} = 2^{-1.6b|I \cup J|}$. But this means that $(\boldsymbol{xy} \mid \boldsymbol{xy} \in R)$ violates 0.8-density on $I \cup J$, which contradicts the maximality of I. □

Claim 11. $|I| \leq O(c/b)$.

Proof. Our gadget is almost balanced: $g^{-1}(1), g^{-1}(0) \geq 2^{2b}/4 = 2^{2(b-1)}$. It follows that $\mathbf{H}_\infty(\boldsymbol{x}_I\boldsymbol{y}_I) \geq 2(b-1)|I|$ for all $I \subseteq [n]$. On the one hand, $\mathbf{H}_\infty(\boldsymbol{x}_I\boldsymbol{y}_I \mid \boldsymbol{xy} \in R) \geq \mathbf{H}_\infty(\boldsymbol{x}_I\boldsymbol{y}_I) - \log(1/\mathbf{Pr}[\,\boldsymbol{xy} \in R\,]) \geq 2(b-1)|I| - c$, where we used the fact that conditioning on an event with probability p lowers the min-entropy by at most $\log(1/p)$. On the other hand, $\mathbf{H}_\infty(\boldsymbol{x}_I\boldsymbol{y}_I \mid \boldsymbol{xy} \in R) < 1.6b|I|$ as $(\boldsymbol{xy} \mid \boldsymbol{xy} \in R)$ violates 0.8-density on I. These two bounds imply the claim. □

In summary, $\boldsymbol{x}'\boldsymbol{y}'$ is fixed on $O(c/b)$ many blocks I, and 0.8-dense on the remaining blocks $\bar{I}$. To apply Lemma 9 we need a pair of independent random variables, and currently $\boldsymbol{x}'$ and $\boldsymbol{y}'$ are highly correlated (e.g., $G(\boldsymbol{x}', \boldsymbol{y}') = z$). Let $\boldsymbol{x}''$ and $\boldsymbol{y}''$ be independent copies of $\boldsymbol{x}'$ and $\boldsymbol{y}'$.

Claim 12. *$\boldsymbol{x}''_{\bar{I}}\boldsymbol{y}''_{\bar{I}}$ is 0.6 dense.*

Proof. Let $J \subseteq \bar{I}$. We want to show $\mathbf{H}_\infty(\boldsymbol{x}''_J\boldsymbol{y}''_J) \geq 1.2b|J|$. We calculate $\mathbf{H}_\infty(\boldsymbol{x}'_J) \geq \mathbf{H}_\infty(\boldsymbol{x}'_J\boldsymbol{y}'_J) - b|J| \geq 1.6b|J| - b|J| = 0.6b|J|$, where the first inequality follows since $b|J|$ is an upper bound on the support size of $\boldsymbol{y}'_J$ (measured in bits) and the second inequality uses the 0.8-density of $\boldsymbol{x}'_{\bar{I}}\boldsymbol{y}'_{\bar{I}}$. The same bound holds for $\mathbf{H}_\infty(\boldsymbol{y}'_J)$. Hence $\mathbf{H}_\infty(\boldsymbol{x}''_J\boldsymbol{y}''_J) = \mathbf{H}_\infty(\boldsymbol{x}''_J) + \mathbf{H}_\infty(\boldsymbol{y}''_J) = \mathbf{H}_\infty(\boldsymbol{x}'_J) + \mathbf{H}_\infty(\boldsymbol{y}'_J) \geq 0.6b|J| + 0.6b|J| = 1.2b|J|$. □

The subrectangle $R' \subseteq R$ we are looking for is now defined as the support of $\boldsymbol{x}''\boldsymbol{y}''$. We can apply Lemma 9 to $\boldsymbol{x}''_{\bar{I}}\boldsymbol{y}''_{\bar{I}}$ to deduce that $G(\boldsymbol{x}'', \boldsymbol{y}'') \in G(R')$ has full support on $\bar{I}$. Moreover, by construction, $G(R')$ is fixed on I and $z \in G(R')$. This concludes the proof of Theorem 4.

Acknowledgements

Thanks to Thomas Watson for many helpful comments on an early draft of this paper. Thanks also to Siu Man Chan, T.S. Jayram, Toniann Pitassi, and Robert Robere for discussions.

References

[Ama14] Kazuyuki Amano. Some improved bounds on communication complexity via new decomposition of cliques. *Discrete Applied Mathematics*, 166(0):249–254, 2014. doi:10.1016/j.dam.2013.09.015.

[BdW02] Harry Buhrman and Ronald de Wolf. Complexity measures and decision tree complexity: a survey. *Theoretical Computer Science*, 288(1):21–43, 2002. doi:10.1016/S0304-3975(01)00144-X.

[BLT14] Nicolas Bousquet, Aurélie Lagoutte, and Stéphan Thomassé. Clique versus independent set. *European Journal of Combinatorics*, 40(0):73–92, 2014. doi:10.1016/j.ejc.2014.02.003.

[CG88] Benny Chor and Oded Goldreich. Unbiased bits from sources of weak randomness and probabilistic communication complexity. *SIAM Journal on Computing*, 17(2):230–261, 1988. doi:10.1137/0217015.

[Die10] Reinhard Diestel. *Graph Theory*. Springer, 4th edition, 2010. URL: http://diestel-graph-theory.com/.

[GLM+14] Mika Göös, Shachar Lovett, Raghu Meka, Thomas Watson, and David Zuckerman. Rectangles are nonnegative juntas. Technical Report TR14-147, Electronic Colloquium on Computational Complexity (ECCC), 2014. URL: http://eccc.hpi-web.de/report/2014/147/.

[HS12] Hao Huang and Benny Sudakov. A counterexample to the Alon–Saks–Seymour conjecture and related problems. *Combinatorica*, 32(2):205–219, 2012. doi:10.1007/s00493-012-2746-4.

[Juk12] Stasys Jukna. *Boolean Function Complexity: Advances and Frontiers*, volume 27 of *Algorithms and Combinatorics*. Springer, 2012.

[KLO99] Eyal Kushilevitz, Nathan Linial, and Rafail Ostrovsky. The linear-array conjecture in communication complexity is false. *Combinatorica*, 19(2):241–254, 1999. doi:10.1007/s004930050054.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity*. Cambridge University Press, 1997.

[KW09a] Eyal Kushilevitz and Enav Weinreb. The communication complexity of set-disjointness with small sets and 0-1 intersection. In *Proceedings of the 50th Symposium on Foundations of Computer Science (FOCS)*, pages 63–72, 2009. doi:10.1109/FOCS.2009.15.

[KW09b] Eyal Kushilevitz and Enav Weinreb. On the complexity of communication complexity. In *Proceedings of the 41st Symposium on Theory of Computing (STOC)*, pages 465–474, 2009. doi:10.1145/1536414.1536479.

[KW14] Volker Kaibel and Stefan Weltge. A short proof that the extension complexity of the correlation polytope grows exponentially. *Discrete & Computational Geometry*, To appear, 2014. doi:10.1007/s00454-014-9655-9.

[NS94] Noam Nisan and Mario Szegedy. On the degree of boolean functions as real polynomials. *Computational Complexity*, 4(4):301–313, 1994. doi:10.1007/BF01263419.

[NW95] Noam Nisan and Avi Wigderson. On rank vs. communication complexity. *Combinatorica*, 15(4):557–565, 1995. doi:10.1007/BF01192527.

[Raz92] Alexander Razborov. On the distributional complexity of disjointness. *Theoretical Computer Science*, 106(2):385–390, 1992. doi:10.1016/0304-3975(92)90260-M.

[SA14] Manami Shigeta and Kazuyuki Amano. Ordered biclique partitions and communication complexity problems. *Discrete Applied Mathematics*, To appear, 2014. doi:10.1016/j.dam.2014.10.029.

[Sav02] Petr Savický. On determinism versus unambiquous nondeterminism for decision trees. Technical Report TR02-009, Electronic Colloquium on Computational Complexity (ECCC), 2002. URL: http://eccc.hpi-web.de/report/2002/009/.

[Vad12] Salil Vadhan. Pseudorandomness. *Foundations and Trends in Theoretical Computer Science*, 7(1–3):1–336, 2012. doi:10.1561/0400000010.

[Yan91] Mihalis Yannakakis. Expressing combinatorial optimization problems by linear programs. *Journal of Computer and System Sciences*, 43(3):441–466, 1991. doi:10.1016/0022-0000(91)90024-Y.

ECCC ISSN 1433-8092

http://eccc.hpi-web.de